

WHAT IS CLAIMED IS:

1 1. An edge counter comprising:
2 an input receiving an input signal and an output
3 on which an output signal is driven; and
4 a set of logic gates between the input and
5 output, the logic gates configured to change a state of the
6 edge counter with each transition of the input signal and
7 to produce an output signal having a cycle corresponding to
8 a predetermined number of transitions of the input signal.

1 2. The edge counter according to claim 1, wherein
2 the predetermined number may be odd or even.

1 3. The edge counter according to claim 1, wherein a
2 signal path between the input and output through the logic
3 gates includes a sequence of only two logic gates.

1 4. The edge counter according to claim 1, wherein
2 the logic gates generate a set of intermediate signals, at
3 least one of the intermediate signals changing state in
4 response to transition of the input signal.

1 5. A wireless receiver including the edge counter
2 according to claim 1, the wireless receiver further
3 comprising one of a local oscillator and a clock divider
4 employing the edge counter.

1 6. A wireless communications system including the
2 wireless receiver according to claim 5, the wireless
3 communications system further a wireless transmitter and a
4 communications path between the transmitter and the
5 receiver.

1 7. A method of designing an edge counter comprising:
2 defining a number of intermediate signals
3 sufficient to count a predetermined number of edges;
4 determining states of the intermediate signals to
5 be generated; and
6 from the determined states, deriving a set of
7 logic gates receiving an input signal, generating the
8 intermediate states in response to transitions in the input
9 signal, and producing an output signal having a cycle
10 corresponding to the predetermined number of edges within
11 the input signal.

1 8. The method according to claim 7, further
2 comprising:
3 inserting gray codes for states of the
4 intermediate signals in a table in a manner corresponding
5 to changes based on input clock signal transitions.

1 9. The method according to claim 8, further
2 comprising:
3 inserting the gray codes in the table to
4 correspond to a transition in the output signal.

1 10. The method according to claim 9, further
2 comprising:

3 identifying rows containing gray codes matching a
4 row value.

1 11. The method according to claim 10, further
2 comprising:

3 generating a Karnaugh map for the states of the
4 intermediate signals corresponding to the identified rows;
5 and

6 designing a set of logic gates to implement the
7 logic function represented by the Karnaugh map.

1 12. The method according to claim 11, further
2 comprising:

3 generating a Karnaugh map for each of the
4 intermediate signals and the output signal.

1 13. The method according to claim 7, further
2 comprising:

3 designing the logic gates to have a two gate
4 delay between the input signal and the output signal.

1 14. An edge counter designed by the steps of:
2 defining a number of intermediate signals
3 sufficient to count a predetermined number of edges;
4 determining states of the intermediate signals to
5 be generated; and
6 from the determined states, deriving a set of
7 logic gates receiving an input signal, generating the
8 intermediate states in response to transitions in the input
9 signal, and producing an output signal having a cycle
10 corresponding to the predetermined number of edges within
11 the input signal.

1 15. The edge counter according to claim 14, further
2 designed by the step of:
3 inserting gray codes for states of the
4 intermediate signals in a table in a manner corresponding
5 to changes based on input clock signal transitions.

1 16. The edge counter according to claim 15, further
2 designed by the step of:
3 inserting the gray codes in the table to
4 correspond to a transition in the output signal.

1 17. The edge counter according to claim 16, further
2 designed by the step of:

3 identifying rows containing gray codes matching a
4 row value.

1 18. The edge counter according to claim 17, further
2 designed by the steps of:

3 generating a Karnaugh map for the states of the
4 intermediate signals corresponding to the identified rows;
5 and

6 designing a set of logic gates to implement the
7 logic function represented by the Karnaugh map.

1 19. The edge counter according to claim 18, further
2 designed by the step of:

3 generating a Karnaugh map for each of the
4 intermediate signals and the output signal.

1 20. The edge counter according to claim 14, further
2 designed by the step of:

3 designing the logic gates to have a two gate
4 delay between the input signal and the output signal.